LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Alexander Engelhardt, Anna Rieger, Achim Tresch, Ulrich Mansmann

Efficient Maximum Likelihood Estimation for Pedigree Data with the Sum-Product Algorithm

Technical Report Number 200, 2016
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Efficient Maximum Likelihood Estimation for Pedigree Data with the Sum-Product Algorithm

Alexander Engelhardt[1], Anna Rieger[1], Achim Tresch[2], Ulrich Mansmann[1]

[1]Institute for Medical Informatics, Biometry and Epidemiology, Ludwig-Maximilians-University, Marchioninistr. 15, 81377 Munich, Germany.

[2]Department of Biology, University of Cologne, Zülpicher Straße 47, 50647 Cologne, Germany.

Keywords: Colorectal cancer, Personalized medicine, Cancer risk prediction, Pedigrees, EM algorithm, Factor graphs, Sum-product algorithm

Abstract

Objective

In this paper, we analyze data sets consisting of pedigrees where the response is the age at onset of colorectal cancer (CRC). The occurrence of familial clusters of CRC suggests the existence of a latent, inheritable risk factor. We aimed to compute the probability of a family possessing this risk factor, as well as the hazard rate increase for these risk factor carriers. Due to the inheritability of this risk factor, the estimation necessitates a costly marginalization of the likelihood.

Methods

We therefore developed an EM algorithm by applying factor graphs and the sum-product algorithm in the E-step, reducing the computational complexity from exponential to linear in the number of family members.

Results

Our algorithm is as precise as a direct likelihood maximization in a simulation study and a real family study on CRC risk. For 250 simulated families of size 19 and 21, the runtime of our algorithm is faster by a factor of 4 and 29, respectively. On the largest family (23 members) in the real data, our algorithm is 6 times faster.

Conclusion

We introduce a flexible and runtime-efficient tool for statistical inference in biomedical event data that opens the door for advanced analyses of pedigree data.

1 Introduction

Colorectal cancer (CRC) is one of the most prevalent cancer diseases in Europe and the United States [1], with men having a younger average age at diagnosis [2]. For a small proportion of CRC cases, genetic predispositions are known [3]. Interestingly,

an additional 15–20% of CRC cases occur in familial clusters [4]. Within these clusters, family members show a higher risk of contracting CRC [5]. The cause for these clusters is unknown but assumed to be a risk factor which may be of genetic or environmental origin.

Since cancer develops earlier in these high-risk families, it is of interest to identify them in advance. Subsequently, health insurances can allow members of high-risk families to join screening programs at an earlier age. In this paper, we therefore develop an efficient risk calculator for CRC, i.e. a method for clinicians to assess the familial risk for a specific family, based on their CRC history.

We look at data consisting of a set of pedigrees, where each person has an inheritable latent variable, the risk factor, that influences its response variable, the age at CRC diagnosis. Assuming an inheritance model and a penetrance model, we aim to estimate two parameters: the a-priori probability p_1 for a founder to carry the risk factor, and the penetrance α, i.e. the multiplicative increase of the hazard rate of an individual that carries the risk factor.

A closely related subject is *complex segregation analysis* (CSA). CSA is a method to evaluate whether pedigree data of affected and unaffected offspring agrees with a Mendelian transmission mode and perform hypothesis tests for different models of inheritance [6]. As opposed to segregation analysis, CSA can go one step further and work with pedigrees of arbitrary structure instead of nuclear families, and quantitative traits as well as qualitative traits [7]. We perform a kind of segregation analysis but do not test for a specific genetic model. In accordance with the argument in Houle *et al.* [8], we employ a phenotype-based approach to study the inheritance mechanisms, because the details of genetic causation of CRC are still unknown and complex, and the assumptions of a genotype-based approach may not hold true.

This problem has been approached in previous work of our group [9]. Since the latent variables are unknown but influence the likelihood, a straightforward estimation procedure has to marginalize the likelihood respective to them. The inheritability of this latent variable means that observations within a family are dependent, and the marginalization can not happen on the level of a single person, but over a whole family. Since each latent variable can assume one of two values, the complexity of computing this sum is $\mathcal{O}(2^D)$, where D is the number of family members.

The runtime of this straightforward optimization over the marginalized likelihood is still reasonable when no family has an excessive number of members. However, the number of possible risk constellations within a family grows two-fold with each new family member. As soon as even one family is sufficiently large, the marginalization quickly becomes unfeasible. In these situations, an alternative approach is needed.

The new aspect in this paper is the implementation of an Expectation-Maximization (EM) algorithm for situations when some families are too large for the marginalization procedure. The E-step is nontrivial because the latent variables within a pedigree are dependent, and a straightforward calculation of the marginal posteriors would again be of exponential runtime. For a linear dependency structure (such as in a Hidden Markov Model), the Baum-Welch algorithm [10] is an efficient method for solving the E-step. In our problem, the data instead shows dependency in a tree structure. This dependency structure necessitates using the sum-product algorithm [11] to obtain the marginalized posterior probabilities for the latent variables in the E-step. A similar approach for the marginalization over hidden variables has been proposed and implemented in [12], yet in the completely different context of single cell time lapse image analysis.

We show that the runtime of our EM algorithm is linear instead of exponential in terms of the pedigree size. We also executed a simulation study to show that our algorithm correctly recovers the specified parameters. Finally, we demonstrate the runtime improvement of our algorithm on a real data set: a family study of CRC cases in Upper Bavaria.

Details on the biological relevance of this analysis will be discussed elsewhere [9].

2 Methods

Nomenclature

The data set is composed of families which are represented as pedigrees (Figure 1a). We call individuals at the top of the pedigree, i.e. with unspecified parents, *founder nodes*, and all other persons *nonfounders*. Individuals without any offspring, i.e. at the bottom of the pedigree, are called *final* individuals.

We denote by t_i the chronological age in years of onset of CRC for each person $i = 1, \ldots, n$, if the corresponding censoring indicator c_i equals 1, and the age at censoring if $c_i = 0$. The gender of an observation is denoted by m_i, which is 1 for males and 0 for females. The observed data for one person is thus $x_i = (t_i, c_i, m_i)$.

Each person also has a latent variable z_i which equals 1 if this person is a risk carrier, and 0 if not. We use $\male_i$ and $\female_i$ to denote the position (i.e. the value of i) of the father and mother of person i. For example, if we have a risk status z_i for a nonfounder i, his father's risk status is $z_{\male_i}$.

We denote the set of all i that are founder nodes by F.

The complete data vectors for all patients are called x and z, respectively.

Penetrance model

For persons where $z_i = 1$, we assume an elevated relative risk of developing CRC, which manifests itself through a hazard rate increased by a multiplicative factor α, the *penetrance* [5]. This parameter is unknown and will be estimated.

We assume a Weibull distribution for t_i. The Weibull hazard rate is given by $h(t) = k\lambda^k t^{k-1}$, with the parameters $k > 0$ and $\lambda > 0$. In our relative risk model, we multiply the hazard rate by α if $z_i = 1$ and, additionally, by β if $m_i = 1$. These factors model the increased relative risk for risk carriers and males, respectively. Our hazard rate for an event (i.e. diagnosis of CRC) is then

$$h(t_i) = k\lambda^k t_i^{k-1} \alpha^{z_i} \beta^{m_i}$$

The survival function is defined by $S(t) = \exp(-\int_0^t h(u)\mathrm{d}u)$. With the additional relative risk factors, this becomes

$$S(t_i) = \exp(-(t_i k)^\lambda \alpha^{z_i} \beta^{m_i})$$

The density for one observation i is composed of the product of the survival function and (for uncensored observations) the hazard rate:

$$f(t_i|z_i) = h(t_i)^{c_i} \cdot S(t_i)$$

The observations x_i are conditionally independent given z_i, and the density of the whole data $f(x|z,\theta)$ can be split up into a product of individual densities: $f(x|z,\theta) = \prod_i f(x_i|z_i,\theta)$.

Heritage model

The *founder prevalence*, i.e. the a-priori probability $\mathbb{P}(Z_i = 1)$ for a founder node to carry the risk factor is called p_1, the second parameter we will estimate. The probability for a nonfounder to be a risk carrier is dependent on its parents' risk statuses and the *inheritance probability* p_H. Our model does not allow for spontaneous mutations to risk carrier. If any one of both parents passes down a risk factor $z_{\male_i} = 1$ or $z_{\female_i} = 1$ with the probability p_H, then the probability for the offspring to be a risk carrier is

$$\tilde{p}_i = \mathbb{P}(Z_i = 1 | z_{\male_i}, z_{\female_i}) = p_H z_{\male_i} + p_H z_{\female_i} - p_H^2 z_{\male_i} z_{\female_i}, \tag{1}$$

We denote $\mathbb{P}(Z_i = 1 | z_{\male_i}, z_{\female_i})$ for nonfounders by $\tilde{p}_i$ to emphasize the distinction from p_1 for founders.

A sensitivity analysis found that varying the value of p_H has a negligible effect on the final parameter estimates [9], and thus we chose $p_H = 0.5$ for all our analyses.

Given a pre-defined inheritance probability p_H and a founder prevalence p_1, the probability for a risk vector Z for the entire dataset becomes

$$\begin{aligned}
\mathbb{P}(z) &= \prod_{i \in F} \mathbb{P}(z_i) \cdot \prod_{i \notin F} \mathbb{P}(z_i | z_{\male_i}, z_{\female_i}) \\
&= \prod_{i \in F} p_1^{z_i} (1 - p_1)^{1 - z_i} \cdot \prod_{i \notin F} \tilde{p}_i^{z_i} (1 - \tilde{p}_i)^{1 - z_i}
\end{aligned}$$

Likelihoods

All Weibull parameters (k, λ) as well as the inheritance probability p_H and the risk increase for males (β) are assumed to be known. We set $k = 4$ and $\lambda = 0.0058$ according to [9], $\beta = 2$ according to [2], and $p_H = 0.5$. The complete likelihood where both x and z are observed, is then

$$L(\theta; x, z) = f(x, z) = f(x|z)\mathbb{P}(z) \tag{2}$$

The two factors $f(x|z)$ and $\mathbb{P}(z)$ were defined in the penetrance model and the inheritance model, respectively. The parameter vector in our model is $\theta = (p_1, \alpha)$.

The complete log-likelihood becomes (derivation in Supplementary Material S1)

$$l(\theta; x, z) = \text{const} + (\sum_{i \in F} z_i) \log p_1 + (|F| - \sum_{i \in F} z_i) \log(1 - p_1) + \sum_{i=1}^{n} c_i z_i \log \alpha - (t_i \lambda)^k \alpha^{z_i} \beta^{m_i} \tag{3}$$

We marginalize the non-reduced form of the complete likelihood to obtain the incomplete likelihood $L(\theta; x)$ [13, Eq. 1.5]:

$$L(\theta; x) = \sum_z L(\theta; x, z) \tag{4}$$

To estimate the parameters p_1 and α, one could use a Nelder-Mead optimization [14] on the marginalized likelihood $L(\theta; x)$. However, for a family of size D, the sum over all z has 2^D elements. Even when splitting the sum up across all families (Supplementary Material S3), the number of summands grows exponentially with increasing family size D. Thus, for large families, the computation of the marginalization within the likelihood evaluation quickly becomes unfeasible.

The EM Algorithm

A common approach for finding maximum likelihood estimates in the presence of latent variables is to make use of the EM algorithm. Resources on the EM algorithm are plentiful, including a short tutorial [15], the seminal paper by Dempster, Laird and Rubin [16], and an entire book [17] devoted to the subject.

In short, the EM algorithm proceeds in a loop over two steps: In the *E-step*, one calculates the expected log-likelihood over the latent variables Z, given the observed data and the current parameter estimates. This problem reduces to computing complete-data sufficient statistics [16]. In the subsequent *M-step*, one then updates the estimates of the parameters, given the new expected sufficient statistics from the E-step.

As a convergence criterium, frequent choices include the size of the relative change of either the log-likelihood or the parameter estimates [18]. If the emphasis lies on compliance with a marginalized optimization, implementing and using the function $B(\theta; \theta^{(t)})$ from Dellaert [15] is the better choice, since this function converges to the true likelihood as one approaches the MLE estimates and thus allows a direct comparability between the two methods. However, evaluating B in each iteration is a costly step. Instead, we use the size of the relative change of the parameter estimates for α and p_1 as a stopping criterion. This criterion is more conservative than using the log-likelihood [18], and we are on the safe side by letting the algorithm run a bit longer than it would have to.

To compute the expected log-likelihood $Q(\theta; \theta^{(t)})$, we introduce the *membership probabilities* [17, p. 43] $T_i^{(t)}$, i.e. the probability for *one* person's risk status Z_i, given the *whole* observed data x (derivation available in Supplementary Material S2):

$$\begin{aligned} T_i^{(t)} &= \mathbb{E}_{Z|x,\theta^{(t)}}(Z_i) \\ &= \mathbb{E}_{Z_i|x,\theta^{(t)}}(Z_i) \\ &= \mathbb{P}(Z_i = 1|x, \theta^{(t)}) \\ &= \sum_z z_i \mathbb{P}(z|x, \theta^{(t)}) \end{aligned} \tag{5}$$

Here, the summation is over all admissible combinations of z_i, i.e. $\mathbb{P}(z) > 0$ and $z_i = 1$. The condition on the entire observed data x and the summation over all z will conveniently reduce to a condition on and summation of only the respective family's data x and z (Supplementary Material S3). The target function Q becomes (cf. Equation 3)

$$\begin{aligned} Q(\theta; \theta^{(t)}) &= \mathbb{E}_{Z|x,\theta^{(t)}}[l(\theta; x, Z)] \\ &= \text{const} + \log(p_1)(\sum_{i \in F} T_i^{(t)}) + \log(1 - p_1)(\sum_{i \in F}(1 - T_i^{(t)})) + \\ &\quad + \sum_{i=1}^{n} T_i^{(t)} c_i \log \alpha - (t_i \lambda)^k \beta^{m_i}(T_i^{(t)} \alpha + (1 - T_i^{(t)})) \end{aligned} \tag{6}$$

The M-step

For the M-step, we maximize $Q(\theta;\theta^{(t)})$ respective to α and p_1 to obtain the new parameter estimates for iteration $t+1$. Once the values of all $T_i^{(t)}$ are known, the maximization of Q with respect to p_1 and α is straightforward and has a closed form solution:

$$\alpha^{(t+1)} = \frac{\sum_{i=1}^{n} c_i T_i^{(t)}}{\sum_{i=1}^{n} T_i^{(t)} (t_i\lambda)^k \beta^{m_i}} \tag{7}$$

$$p_1^{(t+1)} = \frac{\sum_{i\in F} T_i^{(t)}}{|F|} \tag{8}$$

The E-step

It follows from Equations 6 through 8 that, as in the 'standard' examples of the EM algorithm, the E-step conveniently reduces to computing the complete-data sufficient statistics $T_i^{(t)}$. The reason for this simplification is the fact that the log-likelihood is linear in the latent data Z. Computing $Q(\theta;\theta^{(t)})$ thus simplifies to replacing each occuring Z_i by its conditional expectation $T_i^{(t)}$. The M-step then uses these "imputed" values of the latent data Z for the updated parameter estimates.

Marginalization of the joint density

In our setting, the difficulty in computing $T_i^{(t)}$ is that the probability for the risk status of one family member Z_i is conditioned on the observed data x of the *entire family* (Supplementary Material S3). To compute these values, we would have to marginalize over all risk vectors z where $z_i = 1$, i.e. $T_i^{(t)} = \sum_r \mathbb{P}(Z = r|x,\theta^{(t)})$, where r is a valid risk vector (i.e. with $\mathbb{P}(Z = r) > 0$ and with $z_i = 1$). We would end up with the same exponential runtime as in a Nelder-Mead optimization.

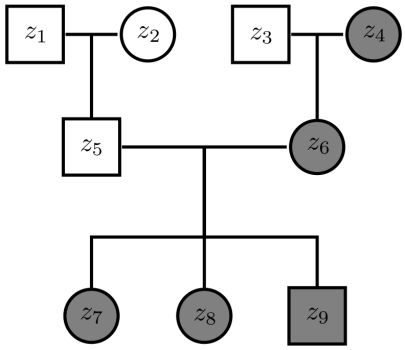
Alternatively, a pedigree can be represented as a Bayesian network [19, 20], also known as a causal probabilistic network (CPN), which in turn can be converted into a factor graph [11]. This representation is advantageous because it allows the efficient computation of marginals via the sum-product algorithm.

The sum-product algorithm [11], also known as the belief propagation algorithm, computes marginalizations of the form of $T_i^{(t)}$ in linear runtime [21, p. 290]. It does this by representing a complex "global" function $g(z)$ – here, $f(x,z|\theta^{(t)})$ – as a factor graph, i.e. a product of multiple "local" functions, $\prod_j \phi_j$, each depending on only a subset of the arguments in $g(z)$.

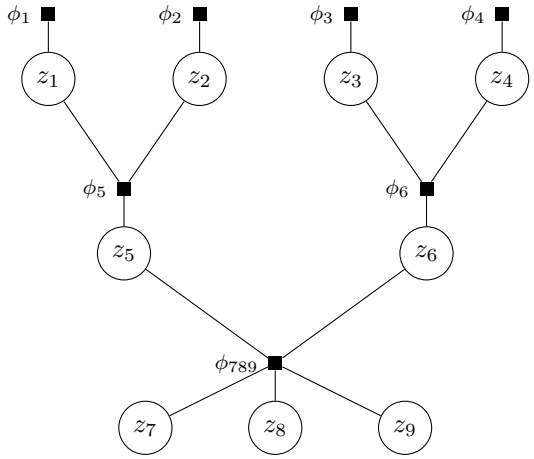
The sum-product algorithm then exploits this structure to efficiently compute marginalizations of $g(z)$ – here, we marginalize the joint density to obtain $f(z_i,x|\theta^{(t)})$. By dividing this joint density through $f(x) = f(Z_i = 1,x|\theta^{(t)}) + f(Z_i = 0,x|\theta^{(t)})$, we ultimately obtain $T_i^{(t)} = \mathbb{P}(Z_i = 1|x,\theta^{(t)})$, which was our actual goal.

Factor graphs

Factor graphs were first introduced by Kschischang [11] to represent factorizations of multivariate functions.



(a) A sample pedigree of a family with 9 members. Squares denote males, circles females. A couple (a connected circle and square in the same row) gives rise to a set of children (the nodes connected to this couple in the row below). Persons shaded in grey are risk carriers. The four grandparents in the top row are the *founder nodes* in this family, the other five persons are *nonfounders*.



(b) A factor graph visualizing the factorization of $g(z) = f(x, z)$ (Equation 2) for the family from Figure 1a. Circles represent variable nodes, and filled squares represent factor nodes, i.e. local functions. The edges show which variables are arguments to which factor. For example, the factor ϕ_6 has three arguments: $\phi_6(z_3, z_4, z_6)$

Figure 1: A pedigree and its corresponding factor graph.

The factor graph in Figure 1b encodes the joint density $g(z) = f(z, x)$ of the family from Figure 1a as the product of 7 factors ϕ_j:

$$g(z) = \phi_1(z_1) \cdot \phi_2(z_2) \cdot \phi_3(z_3) \cdot \phi_4(z_4) \cdot \phi_5(z_1, z_2, z_5) \cdot \phi_6(z_3, z_4, z_6) \cdot \phi_{789}(z_5, z_6, z_7, z_8, z_9) \tag{9}$$

The factors are defined as

$$\phi_J(z_J, z_{\male_J}, z_{\female_J}) = \prod_{j \in J} f(x_j|z_j)\mathbb{P}(z_j|z_{\male_j}, z_{\female_j}),$$

where J is the set of all children with the same parents, which are denoted by $z_{\male_J}$ and $z_{\female_J}$. If ϕ_J is a factor for a founder node, then $z_{\male_J}$ and $z_{\female_J}$ are defined as an empty set and the respective probability $\mathbb{P}(z_j)$ is unconditioned. The exemplary factors for Equation 9 are available in Supplementary Material S4.

The factor ϕ_{789} (Figure 1b) cannot be split up into three factors because the graph edges would then form a *cycle*, which is not allowed, or would necessitate a costly *loopy belief propagation* procedure [11, 22]. Instead, we implement a *clustering* procedure [11] and group the respective densities into one factor per set of parents.

The sum-product algorithm

Having set up a factor graph for each family, we can then apply the sum-product algorithm to compute marginalizations of $f(z, x)$ at each variable node z_i, i.e. $f(z_i, x)$. In our setting, we restrict ourself to family *trees*, i.e. we do not allow for consanguineous marriages (see Figure 2 of Goddard *et al.* [23] for a counterexample), which would again lead to cycles in the corresponding factor graph.

Let $\mu_{z\to\phi}(z)$ denote the message sent from a variable node z to a factor node ϕ, and let $\mu_{\phi\to z}(z)$ denote the message sent from a factor node ϕ to a variable node z. Furthermore, let $n(v)$ denote the set of neighboring nodes of a (factor or variable) node v.

We then define the messages from a variable node to a factor node, and from a factor node to a variable node, as follows [11]:

$$\mu_{z\to\phi}(z) = \prod_{h\in n(z)\setminus\{\phi\}} \mu_{h\to z}(z)$$

$$\mu_{\phi\to z}(z) = \sum_{\sim\{z\}} \left(\phi(Z_\phi) \prod_{y\in n(\phi)\setminus\{z\}} \mu_{y\to\phi}(y) \right)$$

where Z_ϕ is the set of arguments of the factor ϕ. If $h \in n(z) \setminus \{\phi\} = \{\emptyset\}$, e.g. at a variable node of a final individual (the grandchildren z_7, z_8, and z_9 in Figure 1b), the product is defined as 1. The expression $\sum_{\sim\{z\}}$ is adapted from Kschischang et al. [11] and denotes the *not-sum*, i.e. the sum over all variables except z.

Finally, the marginalization, or *termination step*, computes the value of $g_i(z_i) = \sum_{\sim\{z_i\}} g(z)$ as the product of all incoming messages on a variable node z_i. The marginalized $g_i(z_i)$ are equal to $T_i^{(t)}$, i.e. the desired outputs from the E-step in the EM algorithm. We show some example messages and marginalizations of the sum-product algorithm in Supplementary Material S5, and a summarized proof of correct convergence of our algorithm in Supplementary Material S6.

We implemented a sum-product algorithm for computing the marginals of an arbitrary pedigree in R [24] and made it available on GitHub. The code creates one factor graph per family, and therein one factor node per founder, which contains $\phi_i(z_i) = f(x_i|z_i) \cdot \mathbb{P}(z_i)$. Furthermore, we create one factor per set of parents, which contains the product of all densities of all children (but not the parents): $\phi_j(Z_j) = \prod_{i\in K_j} f(x_i|z_i) \cdot \mathbb{P}(z_i|z_{\male_i}, z_{\female_i})$, where Z_j represents all variables within the factor (parents and children), and K_j is the set of children variables connected to ϕ_j.

Messages from a "large" factor containing parents and many children will be summed over all neighboring variable nodes except the destination variable node. By iteratively exploiting the distributive law, this sum can be efficiently broken down from exponential to linear runtime. For an example based on Figure 1b, see Supplemental Material S7.

Application: Estimating the probability of being a risk family

Applying the sum-product algorithm directly implies a straightforward method to compute the probability of being a risk family.

After we have estimated p_1 and α, we can estimate the probability that this family carries the risk factor for a new pedigree (or a pedigree from the original study, i.e. the training data). We define a *risk family* as a family in which at least one member is carrying the risk factor, i.e. $z_i = 1$ for at least one i. This is exactly one minus the probability that *no* family member carries the risk factor. If we restrict the data x and Z to just the family in question, and define the event R as "The family is a risk family", we can then compute

$$\mathbb{P}(R) = 1 - \mathbb{P}(Z = 0|x, \hat{\theta}) \tag{10}$$

$$= 1 - \prod_{i \in F} \mathbb{P}(Z_i = 0|x, \hat{\theta}) \tag{11}$$

$$= 1 - \prod_{i \in F} (1 - T_i^{(t)}) \tag{12}$$

The step from Equation 10 to Equation 11 is possible because the probability that *no family member* carries the risk factor equals the probability that *no founder* carries the risk factor, since the former is true if and only if the latter is true. Then, we can split up the joint probability that no founder carries the risk factor into the individual probabilities $\mathbb{P}(Z_i = 0|x, \hat{\theta})$. This step is possible because we only consider the founders, and their risk probabilities are independent of any other z_i.

Since $\mathbb{P}(Z_i = 0|x, \hat{\theta})$ equals $1 - T_i^{(t)}$ by definition (Equation 5), we can simply run the E-step of the EM algorithm once on the new family to obtain these values. We then multiply over only those $T_i^{(t)}$ where $i \in F$ and obtain $\mathbb{P}(R)$, an estimator for the familial CRC risk.

3 Results

Simulation Study

We performed an *in silico* experiment by simulating data sets with a given p_H, p_1 and α and with a varying number of families (N) and pedigree size (D). The risk status z_i for each founder was randomly sampled with the probability $\mathbb{P}(z_i = 1) = p_1$, the statuses for all nonfounders were sampled according to Equation 1. The age of onset of CRC was then simulated according to a Weibull distribution with the best fitting parameters according to [9], $\lambda = 0.0058$ and $k = 4$, and a risk increase for males of $\beta = 2$:

$$f(t_i|z_i) = h(t_i) \cdot S(t_i) = [k\lambda^k t_i^{k-1} \alpha^{z_i} \beta^{m_i}] \cdot \exp(-(t_i k)^\lambda \alpha^{z_i} \beta^{m_i})$$

We then simulated a censoring age u_i from the following Gaussian distribution: $u_i \sim \mathcal{N}(125, 100)$. The rather optimistic mean censoring age of 125 years was chosen to keep the ratio of censored subjects below 66%, since a higher censoring rate would just necessitate a larger simulated data set to reach the same stability. Each subject's censoring indicator c_i was then set to 1 if $t_i < u_i$ and 0 otherwise. A value of 0 therefore indicates a censored observation. If a subject is censored, t_i was replaced by u_i, the age at censoring.

Runtime improvement

We simulated data sets with different pedigree sizes to investigate the threshold pedigree size from which the EM algorithm is faster than a Nelder-Mead optimization. Figure 2 and Table 1 show the runtime of the Nelder-Mead optimization vs. the EM algorithm for different data sizes and pedigree sizes. The pedigrees used were:

- $D = 5$: Two parents with three children
- $D = 9$: Four grandparents, two parents, three children (Figure 1a)
- $D = 15$: Four generations with only one final individual
- $D = 17$: The same pedigree as for $D = 15$, with one additional parent pair for one founder
- $D = 19$: One more parent pair in the same generation as for $D = 17$
- $D = 21$: One more parent pair in the same generation as for $D = 19$

This suggests that using the EM algorithm is advantageous as soon as *some* families in the data set are large (more than around 17 members). A more advanced EM algorithm could even split the data into small and large pedigrees, and in the E-step use the sum-product algorithm for the larger families, and a "brute force" marginalization for smaller families.

	5	9	15	17	19	21
50	0.01	0.01	0.28	0.75	2.85	10.97
100	0.02	0.02	0.24	0.68	2.62	10.54
150	0.06	0.02	0.33	0.53	2.15	8.33
200	0.01	0.03	0.28	0.57	2.65	12.05
250	0.02	0.03	0.38	0.89	3.89	29.05

Table 1: Runtime ratio (Nelder-Mead over EM algorithm) over different family sizes D (columns) and different number of families N (rows). The EM algorithm is faster for pedigrees of size 19 and above, regardless of the number of families in the data set.

Our algorithm recovers true parameters

We simulated 100 replicated data sets of 500 families of 9 persons as in Figure 1a. In each replication, we chose $p_1 = 0.2$ and $\alpha = 4$ as the parameters and let the Nelder-Mead optimization and the EM algorithm estimate the parameters to investigate their level of agreement. Figure 3 shows scatterplots and Bland-Altman plots to compare the two methods and finds a strong agreement between them. Table 2 shows summary statistics on both methods' parameter estimates in the 100 replications.

The imputation of noninformative parents works

When family members were randomly removed from the data set after simulation, the imputation procedure did not affect the results – both algorithms still converged to the correct result after imputing took place. The simulation and estimation was performed as in Figure 3, but 20 percent of the family members were randomly removed beforehand. The preprocessing

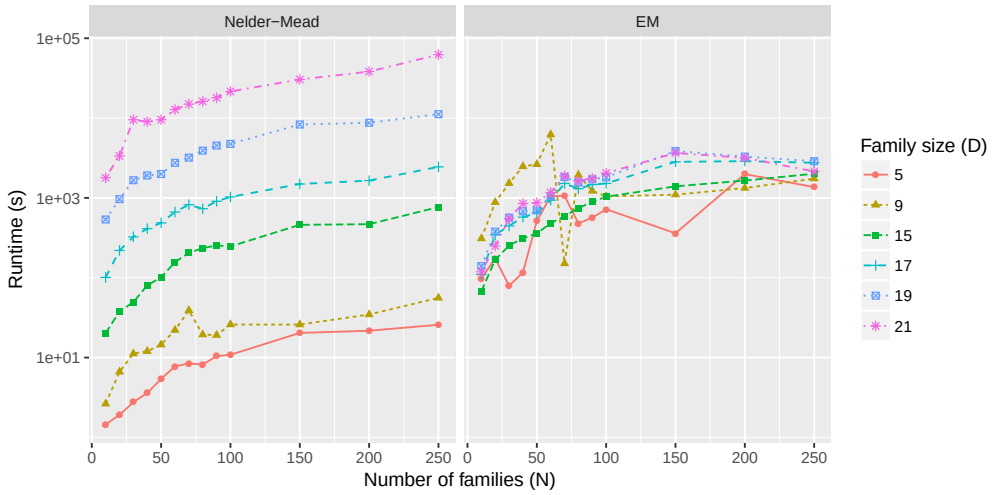


Figure 2: Runtime comparison of Nelder-Mead optimization (left) and the EM algorithm (right). Shown is the runtime in seconds on the y-axis (log-scale) vs. the number of families on the x-axis. The effect of an increasing family size D is negligible with the EM algorithm, but exponential with the Nelder-Mead optimization.

	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
$\hat{p_1}$, N-M	0.1478	0.1740	0.1913	0.1929	0.2078	0.2864
$\hat{p_1}$, EM	0.1476	0.1739	0.1912	0.1929	0.2078	0.2865
$\hat{\alpha}$, N-M	2.888	4.036	4.347	4.310	4.656	5.297
$\hat{\alpha}$, EM	2.890	4.037	4.345	4.310	4.660	5.287

Table 2: Five-point summary and mean values for the parameter estimates of the Nelder-Mead optimization (N-M) and the EM algorithm (EM), based on 100 simulated data sets. The simulation parameters were $p_1 = 0.2$ and $\alpha = 4$.

then performed an imputation of missing members. After our imputing procedure, both algorithms still recover the true parameters (data not shown; reproducible scripts available on GitHub).

Application: Estimating the probability of being a risk family

We computed the posteriori probability of being a CRC risk family for a simulated data set of 1000 pedigrees with 9 persons each, according to Equation 12. The resulting ROC curve is shown in Figure 4. The AUC of 0.74 shows that risk families can be identified with a satisfyingly good rate.

Real Data

We applied our algorithm on a family study of CRC [25]. In this study, patients diagnosed with CRC in the Munich region were recruited. Subsequently, each of these *index patients* was given a questionnaire with a blank pedigree to fill out data

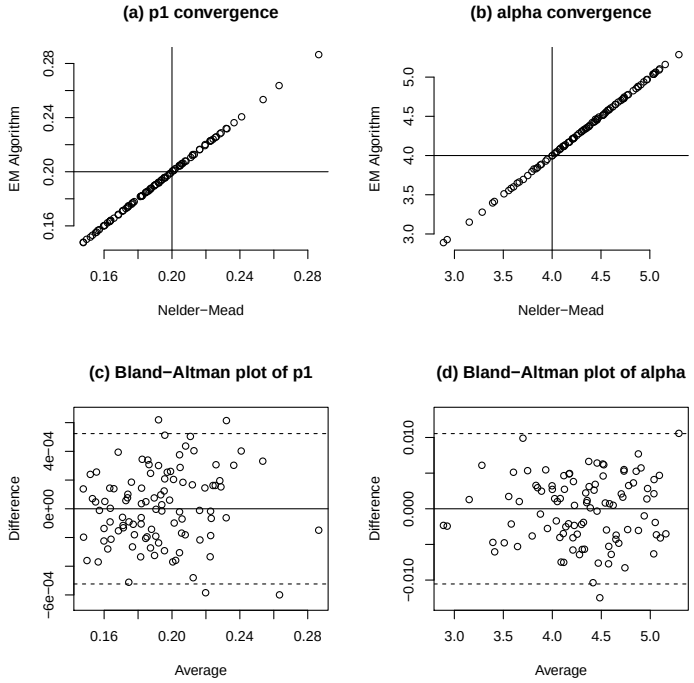


Figure 3: Convergence of 100 replications of simulating and estimating data sets. Each replication used random uniform distributed starting values for $\theta = (p_1, \alpha)$. Figures (a) and (b) show the final parameter estimates for the Nelder-Mead optimization (x-axis) and the EM algorithm (y-axis). Figures (c) and (d) show Bland-Altman plots where the x-axis shows the average of the parameter estimates of the two methods, and the y-axis shows their difference. Horizontal dashed lines are drawn at ± 2 standard deviations of the difference. We see that both estimation methods agree with each other and converge close to the correct result of $p_1 = 0.2$ and $\alpha = 4$ regardless of starting values.

about all known relatives. With this obtained pedigree, the Munich Cancer Registry (MCR) [26] was consulted via an anonymized record-linkage procedure for any CRC diagnoses of the index patient's relatives [27]. The result was a pedigree of family data and CRC diagnoses per index patient. The study was active from September 2012 until June 2014 and resulted in a data set of 611 families, of which 181 were just individuals (a "pedigree" with only one person).

In the real data set, pedigrees were not always recorded in a directly useable manner. For observations with only one available parent, we imputed the missing parent as noninformative ($c_i = 0,\ t_i = 0$ and with the appropriate gender m_i). In cases where a family consisted only of siblings, we imputed both parents as noninformative observations to indicate the relatedness of the siblings. The remaining analysis was analogous to the *in silico* study described in the Methods.

We estimated a prevalence of $p_1 = 0.901$ and a risk factor increase of $\alpha = 5.723$. This rather high a-priori probability may stem from a bias in the data set, since the collection procedure preferrably selected patients and families that are already exposed to risk. A more detailed discussion on the results is given in [9].

The data set contained three families with at least 20 members. For the largest family of 23 persons, using the EM algorithm with the sum-product algorithm instead of a Nelder-Mead optimization showed a reduction of the runtime to 16%.

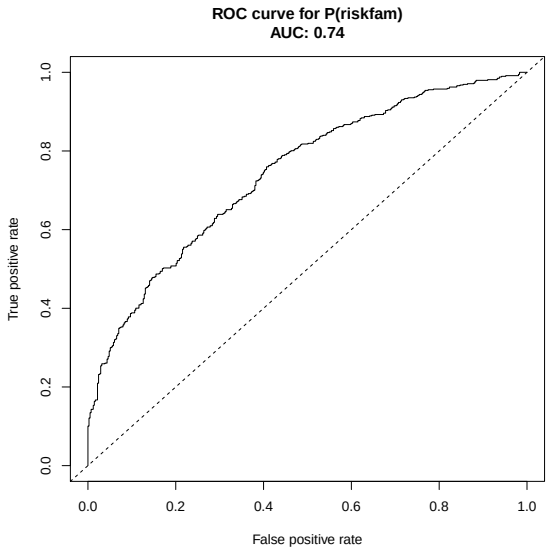


Figure 4: An ROC curve for the probability of being a risk family, based on 1000 simulated families with 9 persons (cf. Figure 1a).

Multiple starts of the EM algorithm are not necessary

We graphed the likelihood surface for both the real data set and several simulated data sets (Figure 5). The likelihood was found to be convex in all settings, showing that there were no local maxima. Thus, starting the EM algorithm from several initial parameter estimates was not necessary. Since there is only one global maximum, running the EM algorithm only once sufficed in all our analyses.

4 Discussion

Directly translating mathematical formulas into computer code often results in a formally correct but slow solution. In our case, a Nelder-Mead optimiziation would need multiple evaluations of the marginalized likelihood, which is unfeasible for larger families. An approach based on *peeling* [28, 29, 30], however, could have been used to reduce the time for evaluating the likelihood. The disadvantage of the Elston-Stewart peeling algorithm is that as soon as the pedigree contains loops, its runtime increases exponentially with the *cutset*, i.e. the number of members that have to be considered jointly [13]. The EM algorithm coupled with the sum-product algorithm can be extended to pedigrees with loops by applying the "loopy belief propagation" procedure [11]. Furthermore, peeling algorithms need to find an optimal peeling order for each pedigree, a problem that still has no gold standard solution today [30]. The sum-product algorithm, on the other hand, directly implies an efficient order of computing the messages, and thus elegantly circumvents this problem. Thompson *et al.* [31] showed that the EM algorithm is a viable alternative to the peeling algorithm in polygenic models. Our approach differs from this in that we skip the detection of responsible genes and instead focus on estimating a family's probability of carrying an (unspecified) CRC risk factor.

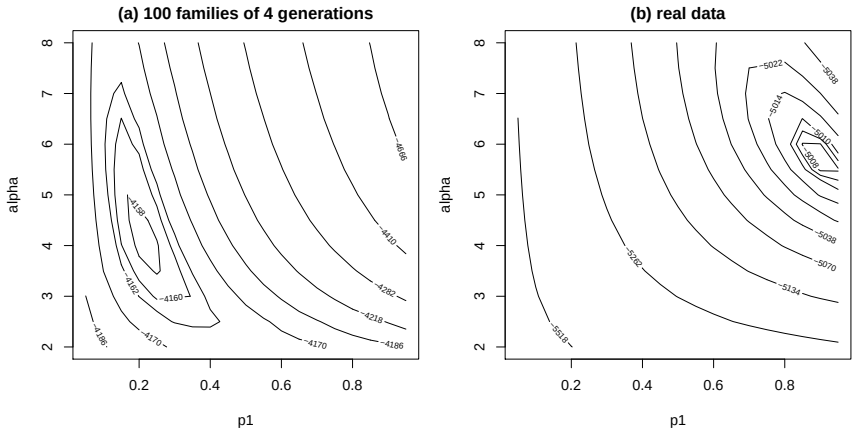


Figure 5: (a) The likelihood surface of 100 pedigrees of size 15 (i.e. 4 generations), with $p_1 = 0.2$ and $\alpha = 4$. (b) The likelihood surface of the real data set. Both graphs show well-behaved, unimodal likelihoods. Further simulations with varying data size and parameters showed similar results (data not shown; reproducible scripts available on GitHub).

The EM algorithm with an approximative Monte Carlo implementation of the E-step has previously been used on pedigrees for segregation analysis [32]. We saw that the EM algorithm in our setting relied on a marginalization over all possible risk vectors for each pedigree. This problem of calculating marginal densities in hierarchical data such as pedigrees has usually been tackled by Markov Chain Monte Carlo (MCMC) simulations and related sampling algorithms [33, 34]. Due to the random sampling, these methods all yield only approximative solutions and may take a long time to reach stable results. We instead use the sum-product algorithm [11] within the EM algorithm to solve the necessary marginalization in the E-step in linear instead of exponential time. This provides a fast and exact solution and allows maximum-likelihood estimation with pedigrees of arbitrary size, that furthermore is not dependent on an arbitrarily chosen number of MCMC simulations.

In contrast to complex segregation analysis, our approach is less specific. In particular, we only model an unspecific risk "component", not necessarily a gene or multiple genes, that is passed down to offspring with a certain pre-defined probability. We chose this phenotype-based approach since the causes for familial occurence of CRC are currently unknown. Our generic model is therefore not fully in line with Mendelian transmission models. Only under the assumption of an autosomal dominant risk factor that is rare, so that an affected individual can be assumed to have the genotype Aa instead of AA, is a constant inheritance probability of $p_H = 0.5$ justifiable. As an advantage, environmental risk factors such as nutrition, lifestyle, or place of residence can be modeled by choosing $p_H = 1$. If one wants to account for a small probability of changing the place of residence, or a probability for children to not adopt the parents' lifestyle, inheritance probabilities less than 1 can be used as well. However, if a large enough data set were available, it should also be possible to estimate p_H robustly enough.

One limitation of the real data set in this study is the relatively small sample size. With around 600 families, the data set was not large enough to obtain stable estimates. However, since the focus of this study is methodological, the data set can still be used to show the runtime improvement of our algorithm.

It should also be noted that for small families, the linear runtime of the sum-product algorithm is *slower* than the exponential runtime of the marginalization, due to the overhead in setting up the factor graph (Figure 2). In our analysis, the sum-product algorithm had significant benefits only as soon as a family consisted of more than 17 members.

Our algorithm can of course also be extended to other models. For example, other penetrance models can be used, such as a "time shift" model, where the hazard rate is not multiplied by a factor α, but instead shifted horizontally, by adding

a "risk advancement" of a specific number of years [35]. It is also possible to use different response distributions besides a Weibull distribution. Furthermore, it is possible to extend our method to model true Mendelian transmission, either by having $Z_i \in \{0, 1, 2\}$ model the number of affected alleles, or by specifying *two* latent variables, $Z_{\male} \in \{0, 1\}$ and $Z_{\female} \in \{0, 1\}$ per individual, one for each allele. One would then need a transmission matrix to specify the probability of each possible outcome for offspring given the statuses of both parents. Ghahramani [36] provides a tutorial on how to extend a Bayesian Network such as a pedigree to deal with multiple latent variables.

When the number of possible genotypes, i.e. the number of possible values for Z_i, increases, both the EM algorithm and the Nelder-Mead optimization suffer from exponential runtime increase. This is the case e.g. when one works with multilocus genotypes. The runtime of the EM algorithm is only linear respective to the family size. However, since the genetic mechanism in our case is unknown, a dichotomized latent variable served our purpose well.

Faster algorithms such as the one presented in this paper also open the door for new analyses that were previously unfeasible. With the sum-product algorithm, we can now conduct large-scale simulation studies for power and sample size determination, and extract further information such as bootstrap confidence intervals from the data.

5 Conclusion

In this paper, we developed an efficient algorithm for maximum likelihood estimation where the observations in the data are partially dependent. The rising size and complexity of modern data sets make it necessary to revisit popular algorithms for data analysis and develop improvements in their efficiency. Here, we considered clinical data in the form of pedigrees, where the presence of latent and inheritable genetic risk factors greatly complicated the analysis procedure. In our case, a standard implementation of the EM algorithm results in a runtime that is still exponential regarding the family sizes, due to the inheritability of the latent variables.

However, by considering the pedigree as a Bayesian network, and then factorizing it with a factor graph and reformulating the E-step by employing a sum-product algorithm, the runtime could be reduced to linear in terms of the family size. Similar to the peeling algorithm [28], the sum-product algorithm in essence breaks down complex pedigrees into *nuclear* families, each consisting of father, mother, and all children. The number of children does not cause exponential growth of runtime because the summation is again broken down between each child.

In conclusion, the combination of an EM algorithm with the sum-product algorithm removes the restrictions that exponential runtime imposes on the analysis due to large families, and opens the door for maximum likelihood estimation on large pedigrees.

As a next step, we plan to make this risk prediction algorithm available as a web interface, so that clinicians can conveniently enter a family's pedigree. It will then aid in assessing familial CRC risk of individual patients.

Code availability

All scripts are available on GitHub at `http://github.com/AlexEngelhardt/sumproduct`.

Conflict of interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The study was approved by the ethical review board of the Medical Faculty of the University of Munich (LMU). The work is supported by the Federal Ministery for Families, Elderly, Women, and Youth (BMFSFJ, FKZ: 3911 401 001).

AE and AT were supported by a BMBF e:Bio grant.

Supplementary Material

This section contains extended derivations of the equations used in this paper.

S1 The complete log-likelihood (Equation 3)

The complete likelihood (Equation 2) becomes

$$\begin{aligned} L(\theta; x, z) = f(x, z) &= \mathbb{P}(z) \cdot f(x|z) \\ &= \prod_{i \in F} \mathbb{P}(z_i) \cdot \prod_{i \notin F} \mathbb{P}(z_i | z_{\male_i}, z_{\female_i}) \cdot \prod_{i=1}^{n} f(x_i|z_i) \\ &= \prod_{i \in F} p_1^{z_i}(1-p_1)^{1-z_i} \cdot \prod_{i \notin F} \tilde{p}_i^{z_i}(1-\tilde{p}_i)^{1-z_i} \cdot \prod_{i=1}^{n} [k\lambda^k t_i^{k-1} \alpha^{z_i} \beta^{m_i}]^{c_i} \exp(-(t_i\lambda)^k \alpha^{z_i} \beta^{m_i}) \end{aligned}$$

Note that the relevant part for α includes all persons, and the part for p_1 only includes the founders. The product over all $i \notin F$ is independent of $\theta = (\alpha, p_1)$ and thus becomes irrelevant in the estimation procedure.

The log-likelihood is then

$$l(\theta; x, z) = (\sum_{i \in F} z_i) \log p_1 + (|F| - \sum_{i \in F} z_i) \log(1 - p_1) + \sum_{i \notin F} [z_i \log \tilde{p}_i + (1 - z_i) \log(1 - \tilde{p}_i)] +$$
$$+ \sum_{i=1}^{n} c_i \cdot [\log k + k \log \lambda + (k-1) \log t_i + z_i \log \alpha + m_i \log \beta] - (t_i \lambda)^k \alpha^{z_i} \beta^{m_i}$$

respective to α and p_1, this reduces to

$$= \text{const} + (\sum_{i \in F} z_i) \log p_1 + (|F| - \sum_{i \in F} z_i) \log(1 - p_1) + \sum_{i=1}^{n} c_i z_i \log \alpha - (t_i \lambda)^k \alpha^{z_i} \beta^{m_i}$$

S2 Derivation of Equation 5

The expected value $\mathbb{E}_{Z|x,\theta^{(t)}}(Z_i)$ is equal to the marginalized expected value $\mathbb{E}_{Z_i|x,\theta^{(t)}}(Z_i)$ because of the following marginalization steps from Z to Z_i:

$$\begin{aligned}
T_i^{(t)} = \mathbb{E}_{Z|x,\theta^{(t)}}(Z_i) &= \sum_z z_i \mathbb{P}(z|x, \theta^{(t)}) \\
&= \sum_{z_1} \cdots \sum_{z_n} z_i \mathbb{P}(z_1, z_2, \ldots, z_n|x, \theta^{(t)}) \\
&= \sum_{z_i} z_i \underbrace{\sum_{z_1} \cdots \sum_{z_{i-1}} \sum_{z_{i+1}} \cdots \sum_{z_n} \mathbb{P}(z_1, z_2, \ldots, z_n|x, \theta^{(t)})}_{=\mathbb{P}(z_i|x,\theta^{(t)})} \\
&= \mathbb{E}_{Z_i|x,\theta^{(t)}}(z_i) \\
&= \mathbb{P}(Z_i = 1|x, \theta^{(t)})
\end{aligned}$$

S3 Efficient computation of likelihoods and marginalizations

Since we assume independence between families, the complete likelihood $L(\theta; x, z)$ can be factorized into a product of N *family likelihoods* [13, Eq. 1.4]. If one denotes the families in a data set by $I = 1, \ldots, N$ and their members by $d = 1, \ldots, D_I$, the index i becomes a combined index I, d from the family index and the member index. We can further define the sub-vectors x_I and z_I to be the observed and latent data for only family I. Note that using this notation, e.g. z_1 is now a vector of risk statuses for family 1, and not the risk status of just the first observation.

Equation 2 then becomes

$$
\begin{aligned}
L(\theta; x, z) &= \prod_{I=1}^{N} f(x_I, z_I) = \prod_{I=1}^{N} \mathbb{P}(z_I) \cdot f(x_I | z_I) \\
&= \prod_{I=1}^{N} \Big\{ \prod_{d \in F_I} \mathbb{P}(z_{I,d}) \cdot \prod_{d \notin F_I} \mathbb{P}(z_{I,d} | z_{\male_{I,d}}, z_{\female_{I,d}}) \cdot \prod_{d=1}^{D_I} f(x_{I,d} | z_{I,d}) \Big\} \\
&= \prod_{I=1}^{N} \Big\{ \prod_{d \in F_I} p_1^{z_{I,d}} (1-p_1)^{1-z_{I,d}} \cdot \prod_{d \notin F_I} \tilde{p}_{I,d}^{z_{I,d}} (1-\tilde{p}_{I,d})^{1-z_{I,d}} \cdot \prod_{d=1}^{D_I} [k \lambda^k t_{I,d}^{k-1} \alpha^{z_{I,d}} \beta^{m_{I,d}}]^{c_{I,d}} \exp(-(t_{I,d}\lambda)^k \alpha^{z_{I,d}} \beta^{m_{I,d}}) \Big\}
\end{aligned}
$$

This notation now allows for computationally elegant marginalizations:

Summation in the marginalized likelihood in Equation 4

Keeping the notation for families and family members, we can rewrite Equation 4 into a more efficient marginalization:

$$
\begin{aligned}
L(\theta; x) &= \sum_{z} L(\theta; x, z) \\
&= \sum_{z} \prod_{I=1}^{N} L(\theta; x_I, z_I) \\
&= \sum_{z_1} \dots \sum_{z_N} L(\theta; x_1, z_1) \cdot \ldots \cdot L(\theta; x_N, z_N) \\
&= \sum_{z_1} L(\theta; x_1, z_1) \cdot \ldots \cdot \sum_{z_N} L(\theta; x_N, z_N) \\
&= \prod_{I=1}^{N} \sum_{z_I} L(\theta; x_I, z_I) \\
&= \prod_{I=1}^{N} \sum_{z_I} f(x_I | z_I, \theta) \; \mathbb{P}(z_I)
\end{aligned}
$$

This way, we do not sum over 2^n possible values for z, but instead $\prod_{I=1}^{N} 2^{D_I}$ values, where D_I is the number of family members in family I.

Marginalizing the risk carrier probability $T_i^{(t)}$ from Equation 5

The marginalization when computing $T_i^{(t)} \equiv T_{I,d}^{(t)}$ can happen more efficiently, summing only over one specific family. Since a $Z_{I,d}$ is independent of all x outside of its respecitve family's x_I, we can replace the condition on x by x_I:

$$
\begin{aligned}
T_{I,d}^{(t)} &= \mathbb{E}_{Z|x,\theta^{(t)}}(Z_{I,d}) \\
&= \mathbb{E}_{Z_{I,d}|x,\theta^{(t)}}(Z_{I,d}) \qquad \text{(Supplementary Material S2)} \\
&= \mathbb{E}_{Z_{I,d}|x_I,\theta^{(t)}}(Z_{I,d}) \\
&= \mathbb{P}(Z_{I,d} = 1|x_I, \theta^{(t)}) \\
&= \sum_{z_I} z_{I,d} \mathbb{P}(z_I|x_I, \theta^{(t)})
\end{aligned}
$$

Therefore, the marginalizations of the factor graph can be efficiently computed for each family separately and combined at the end.

S4 Factor functions ϕ_j for Equation 9

Since $g(z) \equiv f(z, x)$, the factors ϕ_j describe the following functions:

$\phi_1(z_1) = f(x_1|z_1)\ \mathbb{P}(z_1)$

$\phi_2(z_2) = f(x_2|z_2)\ \mathbb{P}(z_2)$

$\phi_3(z_3) = f(x_3|z_3)\ \mathbb{P}(z_3)$

$\phi_4(z_4) = f(x_4|z_4)\ \mathbb{P}(z_4)$

$\phi_5(z_1, z_2, z_5) = f(x_5|z_5)\ \mathbb{P}(z_5|z_1, z_2)$

$\phi_6(z_3, z_4, z_6) = f(x_6|z_6)\ \mathbb{P}(z_6|z_3, z_4)$

$\phi_{789}(z_5, z_6, z_7, z_8, z_9) = f(x_7|z_7)\ \mathbb{P}(z_7|z_5, z_6) \cdot f(x_8|z_8)\ \mathbb{P}(z_8|z_5, z_6) \cdot f(x_9|z_9)\ \mathbb{P}(z_9|z_5, z_6)$.

S5 Some example messages and marginalizations of the sum-product algorithm

We illustrate the sum-product algorithm by calculating two example messages and one example marginalization from the pedigree of Figure 1b.

Firstly, the message $\mu_{z_6 \to \phi_{789}}(z_6)$ from the variable node z_6 to the factor node ϕ_{789} equals

$$\mu_{z_6 \to \phi_{789}}(z_6) = \mu_{\phi_6 \to z_6}(z_6)$$

Secondly, the message $\mu_{\phi_5 \to z_2}(z_2)$ from the factor node ϕ_5 to the variable node z_2 equals

$$\mu_{\phi_5 \to z_2}(z_2) = \sum_{z_1} \sum_{z_5} (\phi_5(z_1, z_2, z_5) \cdot \mu_{z_1 \to \phi_5}(z_1) \mu_{z_5 \to \phi_5}(z_5))$$

Lastly, we compute the example marginalization at the variable node z_5 as

$$g_5(z_5) = f(z_5, x|\theta^{(t)}) = \phi_5(z_5) = \mu_{\phi_5 \to z_5}(z_5) \cdot \mu_{\phi_{789} \to z_5}(z_5)$$

Then,

$$\begin{aligned} T_5^{(t)} &= \mathbb{P}(Z_5 = 1|x, \theta^{(t)}) \\ &= \frac{f(z_5 = 1, x|\theta^{(t)})}{f(z_5 = 0, x|\theta^{(t)}) + f(z_5 = 1, x|\theta^{(t)})} \\ &= \frac{\mu_{\phi_5 \to z_5}(1) \cdot \mu_{\phi_{789} \to z_5}(1)}{\mu_{\phi_5 \to z_5}(0) \cdot \mu_{\phi_{789} \to z_5}(0) + \mu_{\phi_5 \to z_5}(1) \cdot \mu_{\phi_{789} \to z_5}(1)} \end{aligned}$$

This shows that the desired values $T_i^{(t)}$ from the E-step are immediately obtained as soon as all possible messages are computed.

S6 Proof of correct convergence of the EM algorithm

To summarize, we showed that our EM algorithm converges correctly with the following three steps:

(a) The EM algorithm converges to the Maximum Likelihood Estimate (MLE)

(b) The E-step in the EM algorithm for our problem reduces to computing the marginalized expected values $T_i^{(t)} = \mathbb{E}_{Z_i|x,\theta^{(t)}}(Z_i)$

(c) The sum-product algorithm delivers the marginalized densities $T_i^{(t)} = \mathbb{P}(Z_i = 1|x, \theta^{(t)})$.

(a) is clear from [16], and (c) follows from [11]. (b) can be shown as follows:

By definition of the EM algorithm, the E-step consists of computing $Q(\theta; \theta^{(t)}) = \mathbb{E}_{Z|x,\theta^{(t)}}[l(\theta; x, Z)]$. Since the complete log-likelihood $l(\theta; x, z)$ is linear in the latent data z, to obtain $Q(\theta; \theta^{(t)})$ it suffices to replace each Z_i by its conditional expectation given the observed data x and the current fit $\theta^{(t)}$ [17, p. 21].

We can see that $l(\theta; x, z)$ is linear in z from Equation 3. The factor α^{z_i} can be replaced by the equivalent notation $1 + z_i(\alpha - 1)$ because $z_i \in \{0, 1\}$.

A similar approach, called the Baum-Welch algorithm, is used for estimating the parameters of Hidden Markov Models (HMMs) [37]. Since HMMs are a special case of Bayesian Networks such as pedigrees, the same logic applies to our problem [17, cf. p. 73–76].

S7 Evaluating messages in linear time

For example, consider from Figure 1b the message

$$\mu_{\phi_{789}\to z_8}(z_8) = \sum_{z_5=0}^{1}\sum_{z_6=0}^{1}\sum_{z_7=0}^{1}\sum_{z_9=0}^{1} \phi_{789}(z_5,z_6,z_7,z_8,z_9)\cdot \mu_{z_5\to\phi_{789}}(z_5)\mu_{z_6\to\phi_{789}}(z_6)\mu_{z_7\to\phi_{789}}(z_7)\mu_{z_9\to\phi_{789}}(z_9)$$

Since we can decompose $\phi_{789}(z_5,z_6,z_7,z_8,z_9)$ into the product $f(x_7|z_7)\mathbb{P}(z_7|z_5,z_6)\cdot f(x_8|z_8)\mathbb{P}(z_8|z_5,z_6)\cdot f(x_9|z_9)\mathbb{P}(z_9|z_5,z_6)$, the quadruple sum can be split up into

$$\begin{aligned}\mu_{\phi_{789}\to z_8}(z_8) = \sum_{z_5=0}^{1}\sum_{z_6=0}^{1}\Bigg\{ & \mu_{z_5\to\phi_{789}}(z_5)\cdot\mu_{z_6\to\phi_{789}}(z_6)\cdot f(x_8|z_8)\cdot\mathbb{P}(z_8|z_5,z_6) \\ & \cdot\left[\sum_{z_7=0}^{1}\mu_{z_7\to\phi_{789}}(z_7)f(x_7|z_7)\mathbb{P}(z_7|z_5,z_6)\right]\left[\sum_{z_9=0}^{1}\mu_{z_9\to\phi_{789}}(z_9)f(x_9|z_9)\mathbb{P}(z_9|z_5,z_6)\right]\Bigg\}\end{aligned}$$

This representation of the marginalizing sum can now be evaluated in linear time respective to the number of children.

References

[1] Kaatsch P, Spix C, Hentschel S, Katalinic A, Luttmann S, Stegmaier C, Caspritz S, Cernaj J, Ernst A, Folkerts J, et al. Krebs in Deutschland 2009/2010. *Robert Koch-Institut*, 2013.

[2] Kolligs F T, Crispin A, Munte A, Wagner A, Mansmann U, and Göke B. Risk of advanced colorectal neoplasia according to age and gender. *PloS one*, 6(5):e20076, 2011.

[3] Half E, Bercovich D, and Rozen P. Familial adenomatous polyposis. *Orphanet J Rare Dis*, 4(1):22, 2009.

[4] Mendelsohn R B and Markowitz A J. Hereditary colon cancer. *Eur Gastroenterol Hepatol Rev*, 7:251–256, 2011.

[5] Bermejo J L and Hemminki K. Familial risk of cancer shortly after diagnosis of the first familial tumor. *J Natl Cancer Inst*, 97(21):1575–1579, 2005.

[6] Lange K. *Mathematical and Statistical Methods for Genetic Analysis*. Springer, 1997.

[7] Jarvik G P. Complex segregation analyses: uses and limitations. *Am J Hum Genet*, 63(4):942–946, 1998.

[8] Houle D, Govindaraju D R, and Omholt S. Phenomics: the next challenge. *Nat Rev Genet*, 11(12):855–866, 2010.

[9] Rieger A and Mansmann U R. Bayesian prediction of being a colorectal cancer risk family. Manuscript in preparation.

[10] Baum L E. An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[11] Kschischang F R, Frey B J, and Loeliger H A. Factor graphs and the sum-product algorithm. *IEEE Trans Inf Theory*, 47(2):498–519, 2001.

[12] Failmezger H, Dursun E, Schroeder T, Krug A, and Tresch A. Quantification of deterministic and stochastic cell fate components using hidden factor graph models. Submitted to PLoS Comp Biol.

[13] Thompson E A. Statistical inference from genetic data on pedigrees. In *NSF-CBMS regional conference series in probability and statistics*. JSTOR, 2000.

[14] Nelder J A and Mead R. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[15] Dellaert F. The expectation maximization algorithm. *Technical Report*, 2002.

[16] Dempster A P, Laird N M, and Rubin D B. Maximum likelihood from incomplete data via the em algorithm. *J R Stat Soc Series B Stat Methodol*, pages 1–38, 1977.

[17] McLachlan G and Krishnan T. *The EM algorithm and extensions*. John Wiley & Sons, 2007.

[18] Abbi R, El-Darzi E, Vasilakis C, and Millard P. Analysis of stopping criteria for the em algorithm in the context of patient grouping according to length of stay. In *2008 4th International IEEE Conference Intelligent Systems*, volume 1, pages 3–9. IEEE, 2008.

[19] Nielsen T D and Jensen F V. *Bayesian networks and decision graphs*. Springer Science & Business Media, 2009.

[20] Aliferis C F, Tsamardinos I, Statnikov A R, and Brown L E. Causal explorer: A causal probabilistic network learning toolkit for biomedical discovery. In *METMBS*, volume 3, pages 371–376. 2003.

[21] Mezard M and Montanari A. *Information, physics, and computation*. Oxford University Press, 2009.

[22] Ihler A T, John III W F, and Willsky A S. Loopy belief propagation: Convergence and effects of message errors. *J Mach Learn Res*, 6(May):905–936, 2005.

[23] Goddard K, Yu C E, Oshima J, Miki T, Nakura J, Piussan C, Martin G M, Schellenberg G D, and Wijsman E M. Toward localization of the werner syndrome gene by linkage disequilibrium and ancestral haplotyping: lessons learned from analysis of 35 chromosome 8p11. 1-21.1 markers. *Am J Hum Genet*, 58(6):1286, 1996.

[24] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[25] Mansmann U, Stausberg J, Engel J, Heussner P, Birkner B, and Maar C. Familien schützen und stärken – Umgang mit familiärem Darmkrebs. eine Pilotstudie zur Inzidenz von Risikoclustern und zur Möglichkeit ihrer Detektion. *Der Gastroenterologe*, 7:271–272, 2012.

[26] The Munich Cancer Registry. `http://www.tumorregister-muenchen.de/en/index.php`, 2016. [Online; accessed 11-October-2016].

[27] Nasseh D, Engel J, Mansmann U, Tretter W, and Stausberg J. Matching study to registry data: maintaining data privacy in a study on family based colorectal cancer. *Stud Health Technol Inform*, 205:808–812, 2014.

[28] Elston R C and Stewart J. A general model for the genetic analysis of pedigree data. *Hum Hered*, 21(6):523–542, 1971.

[29] Cannings C, Thompson E, and Skolnick M. Probability functions on complex pedigrees. *Adv Appl Probab*, pages 26–61, 1978.

[30] Belonogova N M and Axenovich T I. Optimal peeling order for pedigrees with incomplete genotypic information. *Comp Biol Chem*, 31(3):173–177, 2007.

[31] Thompson E A and Shaw R. Pedigree analysis for quantitative traits: variance components without matrix inversion. *Biometrics*, pages 399–413, 1990.

[32] Guo S W and Thompson E. A monte carlo method for combined segregation and linkage analysis. *Am J Hum Genet*, 51(5):1111, 1992.

[33] Gelfand A E and Smith A F. Sampling-based approaches to calculating marginal densities. *J Am Stat Assoc*, 85(410):398–409, 1990.

[34] Geyer C J and Thompson E A. Constrained monte carlo maximum likelihood for dependent data. *J R Stat Soc Series B Stat Methodol*, pages 657–699, 1992.

[35] Brenner H, Hoffmeister M, and Haug U. Family history and age at initiation of colorectal cancer screening. *Am J Gastroenterol*, 103(9):2326–2331, 2008.

[36] Ghahramani Z. An introduction to hidden markov models and bayesian networks. *Intern J Pattern Recognit Artif Intell*, 15(01):9–42, 2001.

[37] Rabiner L R. A tutorial on hidden markov models and selected applications in speech recognition. *Proc IEEE*, 77(2):257–286, 1989.